COCUS Recruitment Assignment – QA Automation

This is a code challenge for prospective QA Automation Engineers to work at COCUS Portugal.

It is expected that all points are addressed.

This challenge covers:

- Development of a small test automation framework
- Definition of test scenarios
- Behavior Driven Development (BDD)
- RESTful API Testing
- Virtualization using Docker
- Stream processing using SQS on Localstack (https://localstack.cloud)
- AWS command line interface and SDK (https://aws.amazon.com/cli/ and https://aws.amazon.com/tools/)

Deliverable

You are expected to develop a layered test automation framework aimed at testing RESTful APIs and stream processing technologies. The developed framework should include feature files and step definition code. The framework could also include any other layer you deem appropriate (object models, configuration files, wrapper classes, helper classes, etc.)

Expected Goals

- Design your test automation framework in a way generic enough to be able to plug any other kind of test into it.
- You should strive to make the framework as flexible as possible and aim to have tests as easily maintainable as possible.
- Code compiles and all automated tests to execute and pass.
- Push (regularly) all the code to a GitHub repo.

Task 1 – RESTful API tests

Use the provided API at https://reqres.in/ for all tests in this section. All request payloads and responses are in JSON format.

Registration

1. Scenario: Successful registration
    a. POST on https://reqres.in/api/register
    b. Payload: email and password
    c. Response: 200 along with a token
2. Scenario: Unsuccessful registration
    a. POST on https://reqres.in/api/register
    b. Payload: email
    c. Response: 400 along with an error
3. Any other scenarios you would add to this endpoint?

Get User List

1. Scenario: List users
    a. GET on https://reqres.in/api/users
    b. No payload
    c. Response: 200 with list of users

Task 2 – Stream processing test

Setup

- Install Docker virtualization tool on your machine
- Install aws-cli tool on your machine
- Pull an Localstack image
- Have a Docker container running the downloaded Localstack image
- Create a SQS queue on Localstack named “cars”

Test Messages

1. Scenario: Messages are consumed successfully
    a. Produce a few messages on queue with cars details. Message should contain details on a car as follows:
        i. Brand name
        ii. Model
        iii. Number of Doors
        iv. Indicating whether it is a Sports car or not
    b. Consume previously produced messages from queue with cars details.
    c. Compare the produced and consumed messages to verify that all messages were delivered correctly.